Cours de Data Mining – PageRank et HITS

Andreea Dragut

Univ. Aix-Marseille, IUT d'Aix-en-Provence

Plan du cours

- Présentation

Liens en tant que voix (élection)

- première réflexion
  - une page est plus importante si elle a davantage de liens
  - quel type de liens ?
    - liens entrants ?
    - liens sortants ?
- regardons les liens entrants comme des voix (élection) – (PageRank – Google)
  - www.stanford.edu a 23400 liens entrants
  - www.jeannotlapin.com a un lien
- Sont tous les liens entrants de même importance ? – (HITS - IBM)
  - liens depuis des pages importantes comptent davantage
  - question récursive

Formulation récursive simple

- la voix de chaque lien est proportionnelle avec l'importance de la page source
- si la page P avec importance x a n liens sortants, chaque lien reçoit x/n voix
- l'importance de la page P même est la somme des voix de ses liens entrants
- pour l'algorithme PageRank $x = 1$

Formulation matricielle

- la matrice M a une ligne et une colonne pour chaque page web
- supposons que la page j à n liens sortants
  - si $j \longrightarrow i$, alors $M_{i,j} = 1/n$
  - sinon $M_{i,j} = 0$
- M est une matrice stochastique-par-colonne
  - la somme sur chaque colonne vaut 1
- supposons que r est un vecteur avec une entrée par page web
  - r_i est le score de l'importance de la page i
  - appelons ce vecteur le vecteur de rang (rank vector)
  - $|r| = 1$

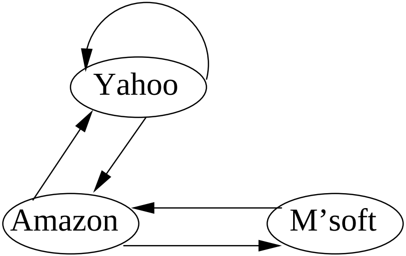
Le modèle "de flot" du web

- les équations de flot peuvent être écrites

$$r = Mr$$

donc le vecteur de rang est un vecteur propre de la matrice
  - en fait, son premier vecteur propre (le vecteur propre principal), correspondant à la valeur propre 1

Exemple



	Y!	A	MS
Y!	1/2	1/2	0
A	1/2	0	1
MS	0	1/2	0

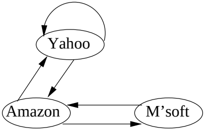
$r = Mr$

$$\begin{aligned} y &= \frac{y}{2} + \frac{a}{2} \\ a &= \frac{y}{2} + m \\ m &= \frac{a}{2} \end{aligned}$$

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

Méthode de l'itération de la puissance

- schéma itératif simple (*relaxation*)
- Supposons qu'il y a N pages web
- Initialisation : $r^0 = [1/N, \ldots, 1/N]^{\mathrm{T}}$
- Itération : $r^{k+1} = Mr^k$
- Condition d'arrêt : lorsque $||r^{k+1} - r^k|| < \varepsilon$
  - $||x||_1 = \sum_{1 \leq i \leq N} |x_i|$ est la norme L_1
  - on peut utiliser toute autre norme (par exemple, euclidienne)



	Y!	A	MS
Y!	1/2	1/2	0
A	1/2	0	1
MS	0	1/2	0

- Itération de la puissance
  - Soit $r^0 = [1/3, 1/3, 1/3]$
  - itérations $r^{k+1} = Mr^k$
- Exemple – vecteur de rang pour $y = \begin{pmatrix} Y! \\ A \\ MS \end{pmatrix}$

$$y = \begin{matrix} 1/3 & 1/3 & 5/12 & 3/8 & & 2/5 \\ 1/3 & 1/2 & 1/3 & 11/24 & \dots & 2/5 \\ 1/3 & 1/6 & 1/4 & 1/6 & & 1/5 \end{matrix}$$

- Pensons à un surfeur web aléatoire (imaginaire)
  - À tout moment t, le surfeur se trouve sur une page quelconque P
  - Au temps $t + 1$, le surfeur suit un lien sortant depuis P, uniformément aléatoirement
  - Le surfeur arrive donc sur une page Q reliée depuis P
  - Le processus se répète indéfiniment
- Soit $p(t)$ un vecteur dont la $i^{\text{ème}}$ composante est la probabilité que le surfeur soit sur la page i au moment t
  - $p(t)$ est une distribution de probabilité sur les pages

La distribution stationaire

- Où est le surfeur au temps $t + 1$?
  - Il suit un lien uniformément aléatoirement
  - $p(t + 1) = Mp(t)$
- Supposons que la marche aléatoire arrive à un état tel que $p(t + 1) = Mp(t) = p(t)$
  - Alors $p(t)$ est appelé distribution stationnaire de la marche aléatoire
- Notre vecteur de rang r satisfait $r = Mr$
  - Il donne donc la distribution stationnaire pour le surfeur aléatoire

Existence et unicité

- Un résultat central de la théorie des marches aléatoires (i.e. processus de Markov) :

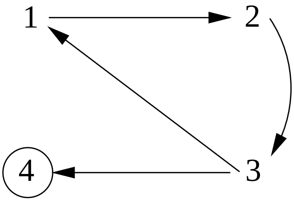
Pour des graphes satisfaisant certaines conditions, la distribution stationnaire est unique et finira par être atteinte quelle que soit la distribution de probabilité initiale au temps $t = 0$.

Composante fortement connexe

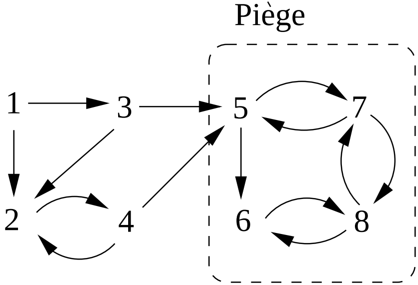
- Dans un graphe orienté (donc là où les sommets sont reliés par des arcs, qui ont une orientation, donc comme les hyperliens entre les pages web), on appelle composante fortement connexe un sous-ensemble de sommets tels que n'importe quels deux sommets sont reliés entre eux.
- En terme de pages web, un groupe de pages web est fortement connexe si de toute page du groupe on peut arriver, en traversant un ou plusieurs liens, à toute autre page du groupe.

Problèmes avec le modèle "de flot" du web

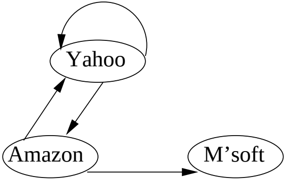
- Impasses
- Certaines pages sont des "impassess" – elles n'ont pas de lien sortant
  - ces pages font que la mesure de l'importance se perde (fuites d'importance)



- Un groupe de pages est un piège à crawlers s'il n'y a pas de lien sortant depuis aucune des pages du groupe vers des pages à l'extérieur du groupe.
  - Un surfeur aléatoire y sera alors piégé
- Les pièges à crawlers ne respectent pas les conditions nécessaires pour le théorème de la marche aléatoire



Piège à crawlers – Impasses

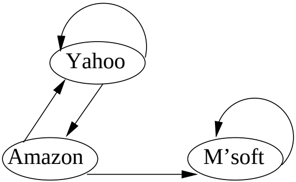


- Itération de la puissance
  - Set $r = [1, 1, 1]$
  - $r^{k+1} = Mr^k$
  - et itérer

	Y!	A	MS
Y!	1/2	1/2	0
A	1/2	0	0
MS	0	1/2	0

- Exemple vecteur de rang pour

$$\begin{array}{c} Y! \\ A \\ MS \end{array} = \begin{array}{cccccccc} 1 & 1 & 1 & 3/4 & 5/8 & & 0 \\ 1 & 1/2 & 1/2 & 1/2 & 3/8 & \dots & 0 \\ 1 & 1/2 & 1/2 & 1/4 & 1/4 & & 0 \end{array}$$



- Itération de la puissance
  - soit $r^0 = [1, 1, 1]$
  - $r^{k+1} = Mr^k$
  - et itérer
  - Exemple

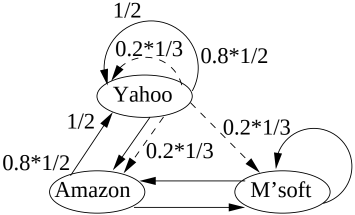
	Y!	A	MS
Y!	1/2	1/2	0
A	1/2	0	0
MS	0	1/2	1

$$\begin{array}{ccccccccc} & & & r^0 & & & & & \\ y = \left(\begin{array}{c} Y! \\ A \\ MS \end{array}\right) & = & & \begin{array}{c} 1 \\ 1 \\ 1 \end{array} & \begin{array}{c} 1 \\ 1/2 \\ 3/2 \end{array} & \begin{array}{c} 1 \\ 1/2 \\ 3/2 \end{array} & \begin{array}{c} 3/4 \\ 1/2 \\ 3/4 \end{array} & \begin{array}{c} 5/8 \\ 3/8 \\ 2 \end{array} & \begin{array}{c} \\ \dots \\ \end{array} \begin{array}{c} 0 \\ 0 \\ 3 \end{array} \end{array}$$

Solution : téléportations aléatoires

- La solution de Google pour les pièges à crawlers
- À chaque pas, le surfeur aléatoire a deux options :
  - Avec probabilité β, suivre un lien aléatoire
  - Avec probabilité $1 - \beta$, sauter à une page de manière uniformément aléatoire
  - Les valeurs usuelles pour β sont entre 0.8 et 0.9
- Le surfeur sera téléporté en dehors du piège après quelques pas

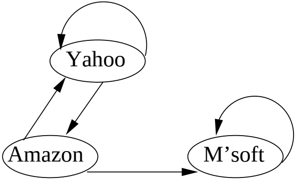
Téléportations aléatoires ($\beta = 0.8$)



$$\begin{array}{cc} & Y! \\ \begin{array}{c} Y! \\ A \\ MS \end{array} & \boxed{\begin{array}{c} 1/2 \\ 1/2 \\ 0 \end{array}} \end{array} \Rightarrow 0.8^* \begin{array}{c} Y! \\ \boxed{\begin{array}{c} 1/2 \\ 1/2 \\ 0 \end{array}} \end{array} + 0.2^* \begin{array}{c} Y! \\ \boxed{\begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array}} \end{array}$$

$$0.8^* \begin{array}{|ccc|} \hline 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \\ \hline \end{array} + 0.2^* \begin{array}{|ccc|} \hline 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ \hline \end{array}$$

$$\begin{array}{c} Y! \\ A \\ MS \end{array} \begin{array}{|ccc|} \hline 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \\ \hline \end{array}$$



$$0.8^* \boxed{\begin{array}{ccc} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{array}} + 0.2^* \boxed{\begin{array}{ccc} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{array}}$$

$$y = \begin{pmatrix} Y! \\ A \\ MS \end{pmatrix}$$

vecteur		1	1.00	0.84	0.776		7/11
de rang	$y =$	1	0.6	0.6	0.536	...	5/11
pour		1	1.4	1.56	1.688		21/11

Ajustement de la matrice pour téléportation depuis une impasse

- Supposons qu'il y a N pages
  - Considérons une pages j, avec un ensemble de liens sortants $O(j)$
  - On a $M_{ij} = 1/|O(j)|$ lorsque $j \rightarrow i$ et $M_{ij} = 0$ autrement
  - La téléportation aléatoire est équivalente
    - au rajout d'un lien de téléportation de j vers toute autre page avec la probabilité $(1 - \beta)/N$
    - à la réduction de la probabilité de suivre chaque lien sortant de $1/|O_{ij}|$ à $\beta/|O_{ij}|$.
    - ou encore, de manière équivalente, à la taxation de chaque page d'une fraction $(1 - \beta)$ de son score, qu'on distribue par la suite uniformément

- Construire la matrice A de dimension $N \times N$ comme suit
  - $A_{ij} = \beta M_{ij} + (1 - \beta)/N$
- Vérifier que A est une matrice stochastique
- Le vecteur r de page rank est le premier vecteur propre de cette matrice
  - il satisfait donc $r = Ar$
- De manière équivalente, r est la distribution stationnaire de la marche aléatoire avec téleportation

Calculer le page rank

- Le pas clé est la multiplication
  - $r^{\text{nouveau}} = Ar^{\text{ancien}}$
- Facile si nous disposons de suffisamment de mémoire vive pour garder en même temps A, r^{ancien} et r^{nouveau}
- Si par exemple $N = 10^9$ pages
  - Nous avons besoin de quatre octets (par exemple) pour chaque entrée
  - Donc $2 \cdot 10^9$ entrées pour les deux vecteurs, presque 8GO
  - La matrice A possède N^2 entrées
    - 10^{18} est un nombre très grand !

Réarranger l'équation

$$
\begin{aligned}
r &= Ar && \text{où} \\
A_{ij} &= \beta M_{ij} + (1-\beta)/N && \text{donc} \\
r_i &= \sum_{1 \le j \le N} A_{ij} r_j \\
r_i &= \sum_{1 \le j \le N} [\beta M_{ij} + (1-\beta)/N]\, r_j \\
&= \beta \sum_{1 \le j \le N} M_{ij} r_j + (1-\beta)/N \sum_{1 \le j \le N} r_j \\
&= \beta \sum_{1 \le j \le N} M_{ij} r_j + (1-\beta)/N && \text{parce que } \; |r| = 1 \\
r &= \beta M r + [(1-\beta)/N])N
\end{aligned}
$$

où $[X]_n$ est un vecteur de dimension N avec toutes les entrées égales à x

Formulation matrice rare (sparse matrix)

- On peut réarranger l'équation du page rank ainsi :
  - $r = \beta M r + [(1-\beta)/N]_N$
  - $[(1-\beta)/N]_N$ est un vecteur de dimension N avec toutes les entrées $(1-\beta)/N$
- M est une matrice rare
  - 10 liens par nœud, donc approximativement $10N$ entrées
- Donc à chaque itération nous avons besoin de
  - Calculer $r^{\text{nouveau}} = \beta M r^{\text{ancien}}$
  - Rajouter une valeur constante de $(1-\beta)/N$ à chaque entrée de r^{nouveau}

Encodage de la matrice rare

- On encode la matrice rare utilisant seulement les entrées non-nulles
  - L'espace nécessaire est proportionnel approx. au nombre de liens
  - e.g. $10N$, ou bien $4 \cdot 10 \cdot 10^9 = 40$GO
  - ceci ne rentre toujours pas en mémoire vive, mais rentrera sur le disque dur

nœud source	degré	nœuds destination
0	3	1,5,7
1	5	17,64,113,117,245
2	2	13,23

Algorithme de base

- Supposons qu'on dispose d'assez de mémoire vive pour y mettre r^{nouveau}, et aussi un peu d'espace de travail
  - on stocke r^{ancien} et la matrice M sur disque

Algorithme de base

- Initialiser $r^{\text{ancien}} = [1/N]_N$
- Itérer :
  - Mise à jour : faire un scan séquentiel de M et r^{ancien} pour mettre à jour r^{nouveau}
  - Écrire r^{nouveau} sur disque en tant que r^{ancien}
  - Toutes les quelques itérations, calculer $|r^{\text{nouveau}} - r^{\text{ancien}}|$ et s'arrêter si c'est inférieur au seuil.
    - besoin de lire les deux vecteurs en mémoire vive

Le pas de la mise à jour

- Initialiser toutes les entrées de r^{nouveau} à $(1-\beta)/N$
- Pour chaque page p (de degré sortant n) :
  - Lire en mémoire vive p, n, $\text{dest}_1, \ldots, \text{dest}_n$, $r^{\text{ancien}}(p)$
  - pour $j = 1 \ldots n$:

$$r^{\text{ancien}}(dest_j) += \beta^* r^{\text{ancien}}(p)/n$$

r^{nouveau}

0
1
2
3
4
5
6

src	degré	destination
0	3	1, 5, 6
1	4	17, 64, 113, 117
2	2	13, 23

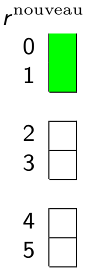
r^{ancien}

0
1
2
3
4
5
6

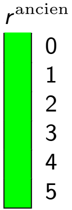
Analyse

- À chaque itération on doit
  - lire r^{ancien} et M
  - écrire r^{nouveau} sur disque
  - Coût E/S : $2|r| + |M|$
- Et si on avait assez de mémoire vive pour y mettre r^{nouveau} et r^{ancien} à la fois ?
- Et si on ne pouvait même pas mettre r^{nouveau} en mémoire vive ?
  - 10 milliards de pages (10^9 pages)

Algorithme de mise-à-jour par blocs



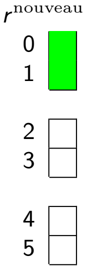
src	degré	destination
0	4	0, 1, 3, 5
1	2	0, 5
2	2	3, 4



Analyse de la mise-à-jour par blocs

- Semblable à la jointure à boucles imbriqués (bases de données)
  - Couper r^{nouveau} en k blocs qui peuvent rentrer en mémoire vive
  - Scanner M et r^{ancien} une fois pour chaque bloc
- k scans de M et r^{ancien}
  - $k(|M| + |r|) + |r| = k|M| + (k+1)|r|$
- Pouvons-nous mieux faire ?
- Indication : M est beaucoup plus grand que r (approx. $10 - 20$ fois), donc on devrait éviter de le lire k fois par itération

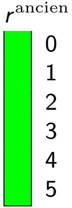
Algorithme de mise-à-jour par blocs-et-rayures



src	degré	destination
0	4	0, 1
1	3	0
2	2	1

src	degré	destination
0	4	3
2	2	3

src	degré	destination
0	4	5
1	3	5
2	2	4



Analyse de la mise-à-jour par blocs-et-rayures

- Découper M en rayures
  - Chaque rayure contient seulement des nœuds destination dans le bloc correspondant de r^{nouveau}
- Un peu de travail supplémentaire par rayure
  - mais d'habitude vallant la peine
- Coût par itération
  - $|M|(1+\varepsilon)+(k+1)|r|$

- Techniques de boosting
  - Techniques pour obtenir une haute pertinence/importance pour une page web
- Techniques de camouflage
  - Techniques pour cacher l'usage du boosting
    - contre les humains et contre les crawlers web

Techniques de boosting

- Spamming avec les termes
  - Manipulation du texte des pages web pour les faire apparaître pertinentes pour les requêtes
- Spamming avec les liens
  - Création de structures de liens qui augmentent le score de PageRank, ou bien le score de hubs et d'autorités

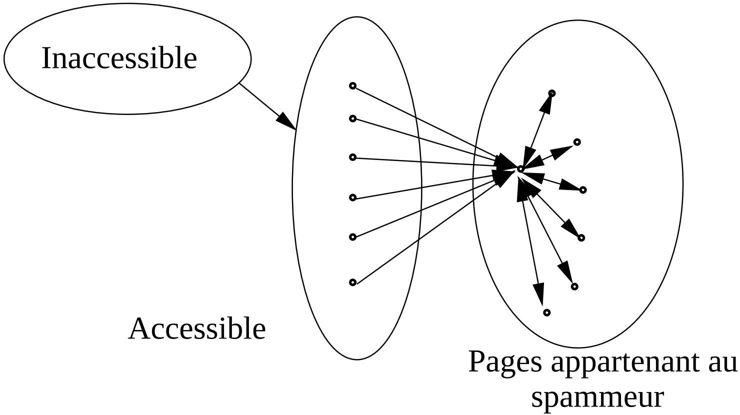
Spamming avec les termes

- Répétition :
  - d'un ou de quelques termes spécifiques (« libre », « pas cher », « viagra »)
  - falsifier les méta-données (titre, auteur)
  - le but est de détourner des schéma de calcul du rang de type TF-IDF
- Dumping
  - d'un grand nombre de termes non-apparentés
  - par exemple : copies entières de dictionnaires
- Tissage
  - copie de pages légitimes et insertion de termes de spam à des positions aléatoires
- Collage de phrases
  - mettre ensemble des propositions et des phrases de sources différentes

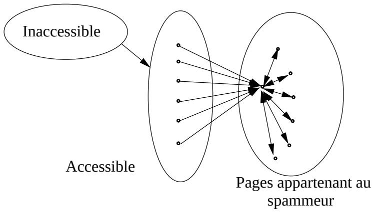
- Du point de vue du spammeur il y a trois types de pages web :
  - pages inaccessibles
  - pages accessibles
    - par exemple, les pages de commentaires et réactions des lecteurs, pour des blogs
    - le spammeur peut y poster des liens vers ses pages
  - Propres pages
    - contrôlées complétement par le spammeur
    - peuvent s'étendre sur plusieurs domaines

Des fermes de liens

- Le but du spammeur :
  - maximiser le score de PageRank de la page cible t
- Technique :
  - obtenir autant de liens que possible depuis les pages accessibles vers la page cible t
  - construire une ferme de liens pour démultiplier l'effet de PageRank



Une des organisations les plus commune et effectives pour une ferme de liens



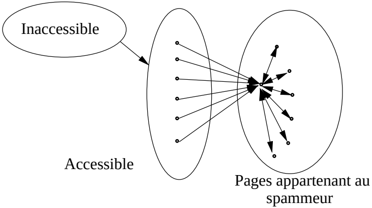
- Notons avec N le nombre de pages du web, et avec M le nombre de pages du spammeur
- Notons le rang contribué par les pages accessible avec x
- Notons le score total de PageRank de la page cible avec y
- Le rang de chaque page « de la ferme » sera $\beta y/M + (1-\beta)/N$

$$\begin{aligned} y &= x + \beta M\left[\beta y/M + (1-\beta)/N\right] + (1-\beta)/N \\ &= x + \beta^2 y + \beta(1-\beta)M/N + \underbrace{(1-\beta)/N}_{\text{très petit -- on l'ignore}} \end{aligned}$$

$$y = x/(1-\beta^2) + cM/N$$

où $c = \beta/(1+\beta)$

Analyse



- $y = x/(1-\beta^2) + cM/N$
  - où $c = \beta/(1+\beta)$
- Pour $\beta = 0.85$, $1/(1-\beta^2) = 3.6$
- Effet de démultiplication pour le score PageRank « acquis »
- En augmentant M, on peut rendre y aussi grand qu'on veut

Détection du spam

- Spam de termes
  - Analysation du texte utilisant les méthodes statistiques
    - par exemple, Bayes naïf, régression logistique
  - Semblable au filtrage de spam du courrier électronique
  - Utile également : détection de pages quasiment dupliquées
- Spam de liens
  - des recherches en cours
  - une approche : TrustRank (« le rang de confiance »)

L'idée du TrustRank

- Principe de base : isolation approximative
  - Il est rare qu'une page qui est « bonne » pointe vers une page qui est « mauvaise » (donc de spam).
- Prenons un ensemble de pages « graines » du web
- Une approche raisonnable, par exemple, aux États-Unis : les `.edu`, `.gov`, `.mil`
- Prenons un oracle (donc un humain) qui étiquette les pages comme étant bonnes ou mauvaises dans l'ensemble de départ (« graines »)
  - tâche très coûteuse
  - on doit donc minimiser la taille de l'ensemble de départ (« graines »)

Propagation de la confiance

- Appeler « les pages de confiance » les pages bonnes dans l'ensemble de départ (« graines »)
- Mettre la valeur de confiance de chaque page de confiance à 1.
- Propager la confiance à travers les liens de la façon suivante
  - Chaque page reçoit une valeur de confiance entre 0 et 1
  - Considérer une valeur seuil et marquer toutes les pages de confiance inférieure à ce seuil comme étant du spam

Règles pour la propagation de la confiance

- Attenuation de la confiance :
  - Le degré de confiance conferré par une page de confiance décroît avec la distance
- Division de la confiance :
  - Plus le nombre de liens sortants d'une page est grand, moins d'examination attentive est accordée à chaque lien par l'auteur de la page
  - La confiance est donc divisée parmi les liens sortants

Modèle simple

- Supposons que la confiance de la page p est t_p
  - Notons O_p l'ensemble de liens sortants
- Pour chaque $q \in O_p$, p accorde la confiance :
  - $\beta t_p / |O_p|$ pour $0 < \beta < 1$
- La confiance est additive
  - La confiance de p est la somme des valeur de confiance des pages des liens entrants
- Similarité avec le Topic-Specific PageRank
  - à un facteur d'échelle près, TrustRank=PageRank avec les pages de confiance en tant qu'ensemble de téléportation

- Dans le modèle TrustRank, on démarre avec les pages bonnes et on propage la confiance
- Point de vue complémentaire :
  - Quelle fraction du score PageRank d'une page vient de pages de spam ?
  - En pratique, on ne connaît pas toutes les pages de spam, donc on doit les estimer

Estimation de la masse de spam

- Soit $r(p)$ le score PageRank de la page p
- Soit $r^+(p)$ le score PageRank de la page p avec de la téléportation dans les « bonnes » pages seulement.
- Alors
$$r^-(p) = r(p) - r^+(p)$$
- La masse de spam de p est $r^-(p)/r(p)$